NAG Library Routine Document

G02MAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G02MAF performs Least Angle Regression (LARS), forward stagewise linear regression or Least Absolute Shrinkage and Selection Operator (LASSO).

2 Specification

```
SUBROUTINE G02MAF (MTYPE, PRED, PREY, N, M, D, LDD, ISX, LISX, Y,          &
                   MNSTEP, IP, NSTEP, B, LDB, FITSUM, ROPT, LROPT,         &
                   IFAIL)
INTEGER            MTYPE, PRED, PREY, N, M, LDD, ISX(LISX), LISX,          &
                   MNSTEP, IP, NSTEP, LDB, LROPT, IFAIL
REAL (KIND=nag_wp) D(LDD,*), Y(N), B(LDB,*), FITSUM(6,MNSTEP+1),          &
                   ROPT(LROPT)
```

3 Description

G02MAF implements the LARS algorithm of Efron *et al.* (2004) as well as the modifications needed to perform forward stagewise linear regression and fit LASSO and positive LASSO models.

Given a vector of n observed values, $y = \{y_i : i = 1, 2, \ldots, n\}$ and an $n \times p$ design matrix X, where the jth column of X, denoted x_j, is a vector of length n representing the jth independent variable x_j, standardized such that $\sum_{i=1}^{n} x_{ij} = 0$, and $\sum_{i=1}^{n} x_{ij}^2 = 1$ and a set of model parameters β to be estimated from the observed values, the LARS algorithm can be summarised as:

1. Set $k = 1$ and all coefficients to zero, that is $\beta = 0$.
2. Find the variable most correlated with y, say x_{j_1}. Add x_{j_1} to the 'most correlated' set $\mathcal{A}$. If $p = 1$ go to 8.
3. Take the largest possible step in the direction of x_{j_1} (i.e., increase the magnitude of β_{j_1}) until some other variable, say x_{j_2}, has the same correlation with the current residual, $y - x_{j_1}\beta_{j_1}$.
4. Increment k and add x_{j_k} to $\mathcal{A}$.
5. If $|\mathcal{A}| = p$ go to 8.
6. Proceed in the 'least angle direction', that is, the direction which is equiangular between all variables in $\mathcal{A}$, altering the magnitude of the parameter estimates of those variables in $\mathcal{A}$, until the kth variable, x_{j_k}, has the same correlation with the current residual.
7. Go to 4.
8. Let $K = k$.

As well as being a model selection process in its own right, with a small number of modifications the LARS algorithm can be used to fit the LASSO model of Tibshirani (1996), a positive LASSO model, where the independent variables enter the model in their defined direction (i.e., $\beta_{kj} \geq 0$), forward stagewise linear regression (Hastie *et al.* (2001)) and forward selection (Weisberg (1985)). Details of the required modifications in each of these cases are given in Efron *et al.* (2004).

The LASSO model of Tibshirani (1996) is given by

$$\underset{\alpha,\beta_k \in \mathbb{R}^p}{\text{minimize}} \left\| y - \alpha - X^{\mathrm{T}} \beta_k \right\|^2 \quad \text{subject to} \quad \|\beta_k\|_1 \leq t_k$$

for all values of t_k, where $\alpha = \bar{y} = n^{-1} \sum_{i=1}^{n} y_i$. The positive LASSO model is the same as the standard LASSO model, given above, with the added constraint that

$$\beta_{kj} \geq 0, \quad j = 1, 2, \ldots, p.$$

Unlike the standard LARS algorithm, when fitting either of the LASSO models, variables can be dropped as well as added to the set $\mathcal{A}$. Therefore the total number of steps K is no longer bounded by p.

Forward stagewise linear regression is an iterative procedure of the form:

1. Initialize $k = 1$ and the vector of residuals $r_0 = y - \alpha$.
2. For each $j = 1, 2, \ldots, p$ calculate $c_j = x_j^{\mathrm{T}} r_{k-1}$. The value c_j is therefore proportional to the correlation between the jth independent variable and the vector of previous residual values, r_k.
3. Calculate $j_k = \underset{j}{\text{argmax}} \left| c_j \right|$, the value of j with the largest absolute value of c_j.
4. If $\left| c_{j_k} \right| < \epsilon$ then go to 7.
5. Update the residual values, with

   $$r_k = r_{k-1} + \delta \ \text{sign}\left(c_{j_k}\right) x_{j_k}$$

   where δ is a small constant and $\text{sign}\left(c_{j_k}\right) = -1$ when $c_{j_k} < 0$ and 1 otherwise.
6. Increment k and go to 2.
7. Set $K = k$.

If the largest possible step were to be taken, that is $\delta = \left| c_{j_k} \right|$ then forward stagewise linear regression reverts to the standard forward selection method as implemented in G02EEF.

The LARS procedure results in K models, one for each step of the fitting process. In order to aid in choosing which is the most suitable Efron *et al.* (2004) introduced a C_p-type statistic given by

$$C_p^{(k)} = \frac{\left\| y - X^{\mathrm{T}} \beta_k \right\|^2}{\sigma^2} - n + 2\nu_k,$$

where ν_k is the approximate degrees of freedom for the kth step and

$$\sigma^2 = \frac{n - y^{\mathrm{T}} y}{\nu_K}.$$

One way of choosing a model is therefore to take the one with the smallest value of $C_p^{(k)}$.

4 References

Efron B, Hastie T, Johnstone I and Tibshirani R (2004) Least Angle Regression *The Annals of Statistics (Volume 32)* **2** 407–499

Hastie T, Tibshirani R and Friedman J (2001) *The Elements of Statistical Learning: Data Mining, Inference and Prediction* Springer (New York)

Tibshirani R (1996) Regression Shrinkage and Selection via the Lasso *Journal of the Royal Statistics Society, Series B (Methodological) (Volume 58)* **1** 267–288

Weisberg S (1985) *Applied Linear Regression* Wiley

5 Arguments

1: MTYPE – INTEGER *Input*

On entry: indicates the type of model to fit.

MTYPE = 1
: LARS is performed.

MTYPE = 2
: Forward linear stagewise regression is performed.

MTYPE = 3
: LASSO model is fit.

MTYPE = 4
: A positive LASSO model is fit.

Constraint: MTYPE = 1, 2, 3 or 4.

2: PRED – INTEGER *Input*

On entry: indicates the type of data preprocessing to perform on the independent variables supplied in D to comply with the standardized form of the design matrix.

PRED = 0
: No preprocessing is performed.

PRED = 1
: Each of the independent variables, x_j, for $j = 1, 2, \ldots, p$, are mean centred prior to fitting the model. The means of the independent variables, $\bar{x}$, are returned in B, with $\bar{x}_j = \mathrm{B}(j, \mathrm{NSTEP} + 2)$, for $j = 1, 2, \ldots, p$.

PRED = 2
: Each independent variable is normalized, with the jth variable scaled by $1/\sqrt{x_j^{\mathrm{T}} x_j}$. The scaling factor used by variable j is returned in $\mathrm{B}(j, \mathrm{NSTEP} + 1)$.

PRED = 3
: As PRED = 1 and 2, all of the independent variables are mean centred prior to being normalized.

Suggested value: PRED = 3.

Constraint: PRED = 0, 1, 2 or 3.

3: PREY – INTEGER *Input*

On entry: indicates the type of data preprocessing to perform on the dependent variable supplied in Y.

PREY = 0
: No preprocessing is performed, this is equivalent to setting $\alpha = 0$.

PREY = 1
: The dependent variable, y, is mean centred prior to fitting the model, so $\alpha = \bar{y}$. Which is equivalent to fitting a non-penalized intercept to the model and the degrees of freedom etc. are adjusted accordingly.

The value of α used is returned in $\mathrm{FITSUM}(1, \mathrm{NSTEP} + 1)$.

Suggested value: PREY = 1.

Constraint: PREY = 0 or 1.

4: N – INTEGER *Input*

On entry: n, the number of observations.

Constraint: $N \geq 1$.

5: M – INTEGER *Input*

On entry: m, the total number of independent variables.

Constraint: $M \geq 1$.

6: D(LDD, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array D must be at least M.

On entry: D, the data, which along with PRED and ISX, defines the design matrix X. The ith observation for the jth variable must be supplied in $D(i,j)$, for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, M$.

7: LDD – INTEGER *Input*

On entry: the first dimension of the array D as declared in the (sub)program from which G02MAF is called.

Constraint: $LDD \geq N$.

8: ISX(LISX) – INTEGER array *Input*

On entry: indicates which independent variables from D will be included in the design matrix, X.

If $LISX = 0$, all variables are included in the design matrix and ISX is not referenced.

If $LISX = M$, for $j = 1, 2, \dots, M$ when $ISX(j)$ must be set as follows:

$ISX(j) = 1$
To indicate that the jth variable, as supplied in D, is included in the design matrix;

$ISX(j) = 0$
To indicated that the jth variable, as supplied in D, is not included in the design matrix;

and $p = \sum_{j=1}^{m} ISX(j)$.

Constraint: if $LISX = M$, $ISX(j) = 0$ or 1 and at least one value of $ISX(j) \neq 0$, for $j = 1, 2, \dots, M$.

9: LISX – INTEGER *Input*

On entry: length of the ISX array.

Constraint: $LISX = 0$ or M.

10: Y(N) – REAL (KIND=nag_wp) array *Input*

On entry: y, the observations on the dependent variable.

11: MNSTEP – INTEGER *Input*

On entry: the maximum number of steps to carry out in the model fitting process.

If $MTYPE = 1$, i.e., a LARS is being performed, the maximum number of steps the algorithm will take is $\min(p, n)$ if $PREY = 0$, otherwise $\min(p, n - 1)$.

If $MTYPE = 2$, i.e., a forward linear stagewise regression is being performed, the maximum number of steps the algorithm will take is likely to be several orders of magnitude more and is no longer bound by p or n.

If MTYPE $= 3$ or 4, i.e., a LASSO or positive LASSO model is being fit, the maximum number of steps the algorithm will take lies somewhere between that of the LARS and forward linear stagewise regression, again it is no longer bound by p or n.

Constraint: MNSTEP ≥ 1.

12: IP – INTEGER *Output*

On exit: p, number of parameter estimates.

If LISX $= 0$, $p =$ M, i.e., the number of variables in D.

Otherwise p is the number of nonzero values in ISX.

13: NSTEP – INTEGER *Output*

On exit: K, the actual number of steps carried out in the model fitting process.

14: B(LDB, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array B must be at least MNSTEP $+ 2$.

On exit: β the parameter estimates, with $\mathrm{B}(j,k) = \beta_{kj}$, the parameter estimate for the jth variable, $j = 1, 2, \ldots, p$ at the kth step of the model fitting process, $k = 1, 2, \ldots, \mathrm{NSTEP}$.

By default, when PRED $= 2$ or 3 the parameter estimates are rescaled prior to being returned. If the parameter estimates are required on the normalized scale, then this can be overridden via ROPT.

The values held in the remaining part of B depend on the type of preprocessing performed.

If PRED $= 0$,

$$\begin{aligned}\mathrm{B}(j, \mathrm{NSTEP}+1) &= 1\\ \mathrm{B}(j, \mathrm{NSTEP}+2) &= 0\end{aligned}$$

If PRED $= 1$,

$$\begin{aligned}\mathrm{B}(j, \mathrm{NSTEP}+1) &= 1\\ \mathrm{B}(j, \mathrm{NSTEP}+2) &= \bar{x}_j\end{aligned}$$

If PRED $= 2$,

$$\begin{aligned}\mathrm{B}(j, \mathrm{NSTEP}+1) &= 1/\sqrt{x_j^{\mathrm{T}} x_j}\\ \mathrm{B}(j, \mathrm{NSTEP}+2) &= 0\end{aligned}$$

If PRED $= 3$,

$$\begin{aligned}\mathrm{B}(j, \mathrm{NSTEP}+1) &= 1/\sqrt{\left(x_j - \bar{x}_j\right)^{\mathrm{T}}\left(x_j - \bar{x}_j\right)}\\ \mathrm{B}(j, \mathrm{NSTEP}+2) &= \bar{x}_j\end{aligned}$$

for $j = 1, 2, \ldots, p$.

15: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which G02MAF is called.

Constraint: LDB $\geq p$, where p is the number of parameter estimates as described in IP.

16: FITSUM$(6, \mathrm{MNSTEP}+1)$ – REAL (KIND=nag_wp) array *Output*

On exit: summaries of the model fitting process. When $k = 1, 2, \ldots, \mathrm{NSTEP}$,

FITSUM$(1, k)$

$\|\beta_k\|_1$, the sum of the absolute values of the parameter estimates for the kth step of the modelling fitting process. If PRED $= 2$ or 3, the scaled parameter estimates are used in the summation.

FITSUM$(2, k)$

RSS_k, the residual sums of squares for the kth step, where $\text{RSS}_k = \|y - X^{\text{T}}\beta_k\|^2$.

FITSUM$(3, k)$

ν_k, approximate degrees of freedom for the kth step.

FITSUM$(4, k)$

$C_p^{(k)}$, a C_p-type statistic for the kth step, where $C_p^{(k)} = \frac{\text{RSS}_k}{\sigma^2} - n + 2\nu_k$.

FITSUM$(5, k)$

$\hat{C}_k$, correlation between the residual at step $k - 1$ and the most correlated variable not yet in the active set $\mathcal{A}$, where the residual at step 0 is y.

FITSUM$(6, k)$

$\hat{\gamma}_k$, the step size used at step k.

In addition

FITSUM$(1, \text{NSTEP} + 1)$

α, with $\alpha = \bar{y}$ if PREY $= 1$ and 0 otherwise.

FITSUM$(2, \text{NSTEP} + 1)$

RSS_0, the residual sums of squares for the null model, where $\text{RSS}_0 = y^{\text{T}}y$ when PREY $= 0$ and $\text{RSS}_0 = (y - \bar{y})^{\text{T}}(y - \bar{y})$ otherwise.

FITSUM$(3, \text{NSTEP} + 1)$

ν_0, the degrees of freedom for the null model, where $\nu_0 = 0$ if PREY $= 0$ and $\nu_0 = 1$ otherwise.

FITSUM$(4, \text{NSTEP} + 1)$

$C_p^{(0)}$, a C_p-type statistic for the null model, where $C_p^{(0)} = \frac{\text{RSS}_0}{\sigma^2} - n + 2\nu_0$.

FITSUM$(5, \text{NSTEP} + 1)$

σ^2, where $\sigma^2 = \frac{n - \text{RSS}_K}{\nu_K}$ and $K = \text{NSTEP}$.

Although the C_p statistics described above are returned when IFAIL $= 112$ they may not be meaningful due to the estimate σ^2 not being based on the saturated model.

17: ROPT(LROPT) – REAL (KIND=nag_wp) array *Input*

On entry: optional parameters to control various aspects of the LARS algorithm.

The default value will be used for ROPT(i) if LROPT $< i$, therefore setting LROPT $= 0$ will use the default values for all optional arguments and ROPT need not be set. The default value will also be used if an invalid value is supplied for a particular argument, for example, setting ROPT$(i) = -1$ will use the default value for argument i.

ROPT(1)

The minimum step size that will be taken.

Default is $100 \times eps$ is used, where eps is the ***machine precision*** returned by X02AJF.

ROPT(2)

General tolerance, used amongst other things, for comparing correlations.

Default is ROPT(1).

ROPT(3)

If set to 1, parameter estimates are rescaled before being returned.

If set to 0, no rescaling is performed.

This argument has no effect when PRED $= 0$ or 1.

Default is for the parameter estimates to be rescaled.

ROPT(4)
: If set to 1, it is assumed that the model contains an intercept during the model fitting process and when calculating the degrees of freedom.

If set to 0, no intercept is assumed.

This has no effect on the amount of preprocessing performed on Y.

Default is to treat the model as having an intercept when PREY $= 1$ and as not having an intercept when PREY $= 0$.

ROPT(5)
: As implemented, the LARS algorithm can either work directly with y and X, or it can work with the cross-product matrices, $X^{\mathrm{T}}y$ and $X^{\mathrm{T}}X$. In most cases it is more efficient to work with the cross-product matrices. This flag allows you direct control over which method is used, however, the default value will usually be the best choice.

If ROPT$(5) = 1$, y and X are worked with directly.

If ROPT$(5) = 0$, the cross-product matrices are used.

Default is 1 when $p \geq 500$ and $n < p$ and 0 otherwise.

Constraints:

ROPT$(1) >$ ***machine precision***;
ROPT$(2) >$ ***machine precision***;
ROPT$(3) = 0$ or 1;
ROPT$(4) = 0$ or 1;
ROPT$(5) = 0$ or 1.

18: LROPT – INTEGER *Input*

On entry: length of the options array ROPT.

Constraint: $0 \leq$ LROPT ≤ 5.

19: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, because for this routine the values of the output arguments may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Note: G02MAF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= 11$

On entry, MTYPE $= \langle value \rangle$.
Constraint: MTYPE $= 1$, 2, 3 or 4.

IFAIL $= 21$

On entry, PRED $= \langle value \rangle$.
Constraint: PRED $= 0$, 1, 2 or 3.

IFAIL $= 31$

On entry, PREY $= \langle value \rangle$.
Constraint: PREY $= 0$ or 1.

IFAIL $= 41$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 1.

IFAIL $= 51$

On entry, M $= \langle value \rangle$.
Constraint: M ≥ 1.

IFAIL $= 71$

On entry, LDD $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LDD $\geq$ N.

IFAIL $= 81$

On entry, ISX($\langle value \rangle$) $= \langle value \rangle$.
Constraint: ISX(i) $= 0$ or 1 for all i.

IFAIL $= 82$

On entry, all values of ISX are zero.
Constraint: at least one value of ISX must be nonzero.

IFAIL $= 91$

On entry, LISX $= \langle value \rangle$ and M $= \langle value \rangle$.
Constraint: LISX $= 0$ or M.

IFAIL $= 111$

On entry, MNSTEP $= \langle value \rangle$.
Constraint: MNSTEP ≥ 1.

IFAIL $= 112$

Fitting process did not finish in MNSTEP steps. Try increasing the size of MNSTEP and supplying larger output arrays.
All output is returned as documented, up to step MNSTEP, however, σ and the C_p statistics may not be meaningful.

IFAIL $= 151$

On entry, LDB $= \langle value \rangle$ and M $= \langle value \rangle$.
Constraint: if LISX $= 0$ then LDB $\geq$ M.

IFAIL $= 152$

On entry, LDB $= \langle value \rangle$ and $p = \langle value \rangle$.
Constraint: if LISX $=$ M then LDB $\geq p$.

IFAIL $= 161$

σ^2 is approximately zero and hence the C_p-type criterion cannot be calculated. All other output is returned as documented.

IFAIL $= 162$

$\nu_K = n$, therefore σ has been set to a large value. Output is returned as documented.

IFAIL $= 163$

Degenerate model, no variables added and NSTEP $= 0$. Output is returned as documented.

IFAIL $= 181$

On entry, LROPT $= \langle value \rangle$.
Constraint: $0 \leq$ LROPT ≤ 5.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G02MAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02MAF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

G02MAF returns the parameter estimates at various points along the solution path of a LARS, LASSO or stagewise regression analysis. If the solution is required at a different set of points, for example when performing cross-validation, then G02MCF can be used.

For datasets with a large number of observations, n, it may be impractical to store the full X matrix in memory in one go. In such instances the cross-product matrices $X^{\mathrm{T}}y$ and $X^{\mathrm{T}}X$ can be calculated, using for example, multiple calls to G02BUF and G02BZF, and G02MBF called to perform the analysis.

The amount of workspace used by G02MAF depends on whether the cross-product matrices are being used internally (as controlled by ROPT). If the cross-product matrices are being used then G02MAF internally allocates approximately $2p^2 + 4p + \max(np)$ elements of real storage compared to $p^2 + 3p + \max(np) + 2n + n \times p$ elements when X and y are used directly. In both cases approximately $5p$ elements of integer storage are also used. If a forward linear stagewise analysis is performed than an additional $p^2 + 5p$ elements of real storage are required.

10 Example

This example performs a LARS on a simulated dataset with 20 observations and 6 independent variables.

10.1 Program Text

```
    Program g02mafe

!     G02MAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g02maf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                 :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                            :: i, ifail, ip, k, ldb, ldd, lisx,     &
                                            lropt, m, mnstep, mtype, n, nstep,   &
                                            pred, prey
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable    :: b(:,:), d(:,:), fitsum(:,:),         &
                                            ropt(:), y(:)
      Integer, Allocatable               :: isx(:)
!     .. Intrinsic Procedures ..
      Intrinsic                          :: count, floor, max, repeat
!     .. Executable Statements ..
      Write (nout,*) 'G02MAF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size
      Read (nin,*) n, m

!     Read in the model specification
      Read (nin,*) mtype, pred, prey, mnstep, lisx

!     Read in variable inclusion flags (if specified) and calculate IP
      Allocate (isx(lisx))
      If (lisx==m) Then
        Read (nin,*) isx(1:lisx)
        ip = count(isx(1:m)==1)
      Else
        ip = m
      End If

!     Optional arguments (using defaults)
      lropt = 0
      Allocate (ropt(lropt))

!     Read in the data
      ldd = n
      Allocate (y(n),d(ldd,m))
      Read (nin,*)(d(i,1:m),y(i),i=1,n)

!     Allocate output arrays
      ldb = ip
      Allocate (b(ldb,mnstep+2),fitsum(6,mnstep+1))
```

```
!     Call the model fitting routine
      ifail = -1
      Call g02maf(mtype,pred,prey,n,m,d,ldd,isx,lisx,y,mnstep,ip,nstep,b,ldb,  &
        fitsum,ropt,lropt,ifail)
      If (ifail/=0) Then
        If (ifail/=112 .And. ifail/=161 .And. ifail/=162 .And. ifail/=163)     &
          Then
!         IFAIL = 112, 161, 162 and 163 are warnings, so no need to terminate
!         if they occur
          Go To 100
        End If
      End If

!     Display the parameter estimates
      Write (nout,*) ' Step ', repeat(' ',max((ip-2),0)*5),                    &
        ' Parameter Estimate'
      Write (nout,*) repeat('-',5+ip*10)
      Do k = 1, nstep
        Write (nout,99998) k, b(1:ip,k)
      End Do
      Write (nout,*)
      Write (nout,99999) 'alpha: ', fitsum(1,nstep+1)
      Write (nout,*)
      Write (nout,*)                                                           &
        ' Step     Sum      RSS       df       Cp       Ck     Step Size'
      Write (nout,*) repeat('-',64)
      Do k = 1, nstep
        Write (nout,99997) k, fitsum(1:2,k), floor(fitsum(3,k)+0.5_nag_wp),    &
          fitsum(4:6,k)
      End Do
      Write (nout,*)
      Write (nout,99999) 'sigma^2: ', fitsum(5,nstep+1)

100   Continue
99999 Format (1X,A,F9.3)
99998 Format (2X,I3,10(1X,F9.3))
99997 Format (2X,I3,2(1X,F9.3),1X,I6,1X,3(1X,F9.3))
    End Program g02mafe
```

10.2 Program Data

```
G02MAF Example Program Data
20 6                          :: N,M
1 3 1 6 0                     :: MTYPE,PRED,PREY,MNSTEP,LISX
10.28  1.77  9.69 15.58  8.23 10.44    -46.47
 9.08  8.99 11.53  6.57 15.89 12.58    -35.80
17.98 13.10  1.04 10.45 10.12 16.68   -129.22
14.82 13.79 12.23  7.00  8.14  7.79    -42.44
17.53  9.41  6.24  3.75 13.12 17.08    -73.51
 7.78 10.38  9.83  2.58 10.13  4.25    -26.61
11.95 21.71  8.83 11.00 12.59 10.52    -63.90
14.60 10.09 -2.70  9.89 14.67  6.49    -76.73
 3.63  9.07 12.59 14.09  9.06  8.19    -32.64
 6.35  9.79  9.40 12.79  8.38 16.79    -83.29
 4.66  3.55 16.82 13.83 21.39 13.88    -16.31
 8.32 14.04 17.17  7.93  7.39 -1.09     -5.82
10.86 13.68  5.75 10.44 10.36 10.06    -47.75
 4.76  4.92 17.83  2.90  7.58 11.97     18.38
 5.05 10.41  9.89  9.04  7.90 13.12    -54.71
 5.41  9.32  5.27 15.53  5.06 19.84    -55.62
 9.77  2.37  9.54 20.23  9.33  8.82    -45.28
14.28  4.34 14.23 14.95 18.16 11.03    -22.76
10.17  6.80  3.17  8.57 16.07 15.93   -104.32
 5.39  2.67  6.37 13.56 10.68  7.35    -55.94 :: End of D, Y
```

10.3 Program Results

```
 G02MAF Example Program Results

  Step                         Parameter Estimate
 -----------------------------------------------------------------
    1      0.000      0.000      3.125      0.000      0.000      0.000
    2      0.000      0.000      3.792      0.000      0.000     -0.713
    3     -0.446      0.000      3.998      0.000      0.000     -1.151
    4     -0.628     -0.295      4.098      0.000      0.000     -1.466
    5     -1.060     -1.056      4.110     -0.864      0.000     -1.948
    6     -1.073     -1.132      4.118     -0.935     -0.059     -1.981

 alpha:    -50.037

  Step      Sum       RSS        df        Cp        Ck      Step Size
 ----------------------------------------------------------------
    1     72.446   8929.855      2       13.355   123.227     72.446
    2    103.385   6404.701      3        7.054    50.781     24.841
    3    126.243   5258.247      4        5.286    30.836     16.225
    4    145.277   4657.051      5        5.309    19.319     11.587
    5    198.223   3959.401      6        5.016    12.266     24.520
    6    203.529   3954.571      7        7.000     0.910      2.198

 sigma^2:    304.198
```

This example plot shows the regression coefficients (β_k) plotted against the scaled absolute sum of the parameter estimates ($\|\beta_k\|_1$).